

Global Anomalies



[Contents](#) [Next](#)

Goal: The goal for this tutorial is to show how to extract several AMIP model data, generate global anomalies data and save it to a NetCDF file, and create a global anomalies plot.

We will learn how to loop through a subset of the [AMIP](#) data and extract the specified variable, calculate annual cycle and gridpoint anomalies and generate a global anomaly time series plot and output NetCDF file with this anomaly time series data.

You can download the python script file [global_anomalies.py](#).

Note: We assume that you have an access to the data through the directory 'pcmdi/AMIP3/amip/mo/'.

First let's import all the needed modules

```
import cdms, cdutil, MA, vcs, cdtime  
import string, Numeric, time, sys
```

define the variable name we are going to extract from the data

```
var='tas'
```

define the models for which we will extract the data

```
model=['bmrc-01a','bmrc-90a','bmrc-95a','cccma-90a',  
       'ccsr-95a','ccsr-98a','cnrm-00a','cnrm-95a',  
       'dnm-91a','dnm-95a','dnm-98a','ecmwf-90a','ecmwf-98a',  
       'ecmwf-98b','gfdl-92a','ncep-92a','ncep-99a',  
       'ncep-99b','ncar-03a','ncar-03c','ncar-03d','ukmo-98a',  
       'yonu-01a']
```

set up a description string for addition to the global attributes in the output netcdf

```
model_description=''
```

Loop over all models, open the appropriate model's data with surface temperature, variable name 'tas', check and print the model and the data's shape, and compose the model_description string with the names of all the models:

```
for i in range(0,len(model)):  
    a = cdms.open('/pcmdi/AMIP3/amip/mo/' +  
                 var + '/' + model[i] + '/' + var + '_' + model[i] + '.xml')  
    data = a[var]  
    print i, model[i], data.shape  
    a.close()  
    dm = str(i) + ' = ' + model[i]  
    model_description = model_description + ', ' + dm
```

You'll see the output as follows:

0 bmrc-01a (215, 1, 72, 144)

```

1 bmrc-90a (120, 1, 80, 96)
2 bmrc-95a (120, 1, 80, 96)
3 ccma-90a (120, 1, 48, 96)
4 ccsr-95a (120, 1, 32, 64)
5 ccsr-98a (206, 1, 64, 128)
6 cnrm-00a (236, 1, 64, 128)
7 cnrm-95a (120, 1, 64, 128)
8 dnm-91a (120, 1, 45, 72)
9 dnm-95a (120, 1, 45, 72)
10 dnm-98a (206, 1, 45, 72)
11 ecmwf-90a (120, 1, 64, 128)
12 ecmwf-98a (237, 1, 91, 180)
13 ecmwf-98b (242, 1, 91, 180)
14 gfdl-92a (120, 1, 80, 96)
15 ncep-92a (120, 1, 64, 128)
16 ncep-99a (240, 1, 64, 128)
17 ncep-99b (240, 1, 94, 192)
18 ncar-03a (204, 64, 128)
19 ncar-03c (204, 128, 256)
20 ncar-03d (204, 64, 128)
21 ukmo-98a (206, 1, 73, 96)
22 yonu-01a (206, 1, 46, 72)

```

set up an output array for the global time series

```
glan=MA.zeros([len(model),120],MA.Float)
```

Loop over the files and read data into memory. Subtract the average annual cycle and area-average the departure maps for a global departure/anomaly time series.

```

start_time = cdtime.comptime(1979)
end_time   = cdtime.comptime(1988)
for i in range(0,len(model)):
    a=cdms.open('/pcmdi/AMIP3/amip/mo/' + var+'/'+model[i]+'/'+var+'_'+model[i]+'.xml'
    data=a(var,time=slice(0,120),squeeze=1)
    ac=cdutil.ANNUALCYCLE.climatology(data,time=(start_time, end_time, 'cob'
    data_an=cdutil.ANNUALCYCLE.departures(data,ref=ac)
    print i,model[i],data.shape, data_an.shape
    glan[i,:]=cdutil.averager(data_an,axis='xy')
```

The output will look like that:

```

0 bmrc-01a (120, 72, 144) (120, 72, 144)
1 bmrc-90a (120, 80, 96) (120, 80, 96)
2 bmrc-95a (120, 80, 96) (120, 80, 96)
3 ccma-90a (120, 48, 96) (120, 48, 96)
4 ccsr-95a (120, 32, 64) (120, 32, 64)
5 ccsr-98a (120, 64, 128) (120, 64, 128)
6 cnrm-00a (120, 64, 128) (120, 64, 128)
7 cnrm-95a (120, 64, 128) (120, 64, 128)
8 dnm-91a (120, 45, 72) (120, 45, 72)
9 dnm-95a (120, 45, 72) (120, 45, 72)
10 dnm-98a (120, 45, 72) (120, 45, 72)

```

- 11 ecmwf-90a (120, 64, 128) (120, 64, 128)
- 12 ecmwf-98a (120, 91, 180) (120, 91, 180)
- 13 ecmwf-98b (120, 91, 180) (120, 91, 180)
- 14 gfdl-92a (120, 80, 96) (120, 80, 96)
- 15 ncep-92a (120, 64, 128) (120, 64, 128)
- 16 ncep-99a (120, 64, 128) (120, 64, 128)
- 17 ncep-99b (120, 94, 192) (120, 94, 192)

setup metadata and write out to a netcdf file

```
tim = data.getTime()
runs = Numeric.arange(0, len(model))
runs = cdms.createAxis(runs, id='models')
glan = cdms.createVariable(glan, axes=(runs, tim), id='global_'+var+'_anomalies')
```

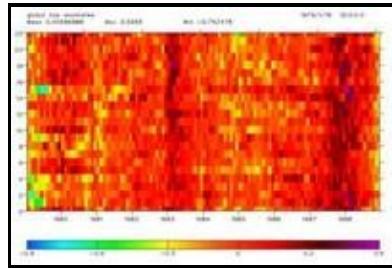
open the NetCDF output file (in your current directory) and write the data ('global_anomalies.nc' ~26kB)

```
q = cdms.open('global_anomalies.nc', 'w')
q.model_designation = model_description
q.write(glan)
q.close()
```

Make a simple time series plot of a global anomaly

```
x = vcs.init()
x.setcolormap('default')
x.plot(glan)
```

Here is the final plot:



This tutorial was provided by [Jay Hnilo](#)

